**Московский государственный технический**

**университет им. Н. Э. Баумана**

**Факультет «Информатика и системы управления»**

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №3

Группа: ИУ5-52Б

Студент: Брусникина М.И.

Преподаватель: Гапанюк Ю.Е.

Москва, 2019 г.

**Задание**

Необходимо разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
   1. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь - <https://docs.python.org/3/library/abc.html>
   2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь - <https://docs.python.org/3/library/functions.html#property>
   3. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
   4. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math - <https://docs.python.org/3/library/math.html>
   5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
   6. Для классов «Прямоугольник», «Квадрат», «Круг»:
      1. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
      2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
4. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - <https://docs.python.org/3/library/__main__.html>). Создайте следующие объекты и выведите о них информацию в консоль:
   1. Прямоугольник синего цвета шириной 3 и высотой 2.
   2. Круг зеленого цвета радиусом 5.
   3. Квадрат красного цвета со стороной 5.

**Исходный код**

**Файл \_\_main\_\_.py**

from Rectangle import Rectangle

from Square import Square

from Circle import Circle

if \_\_name\_\_ == "\_\_main\_\_":

rec = Rectangle(3, 2, 'blue')

cir = Circle(5, 'green')

sq = Square(5, 'red')

print(rec.\_\_repr\_\_())

print(cir.\_\_repr\_\_())

print(sq.\_\_repr\_\_())

**Файл Figure.py**

# Абстрактный класс "Геометрическая фигура"

from abc import ABC, abstractmethod

class Figure(ABC):

@abstractmethod

def square(self):

pass

**Файл Color.py**

# Класс "Цвет фигуры"

class Color:

def \_\_init\_\_(self):

self.\_color = None

@property

def colorprop(self):

return self.\_color

@colorprop.setter

def colorprop(self, new\_color):

self.\_color = new\_color

**Файл Rectangle.py**

# Класс "Прямоугольник"

from Color import Color

from Figure import Figure

class Rectangle(Figure):

FIGURE\_TYPE = "Прямоугольник"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, width, height, color):

self.width = width

self.height = height

self.color = Color()

self.color.colorprop = color

def square(self):

return self.width \* self.height

def \_\_repr\_\_(self):

return '{} {} цвета с шириной {}, высотой {} и площадью {}'.format(

Rectangle.get\_figure\_type(),

self.color.colorprop,

self.height,

self.width,

self.square()

)

**Файл Square.py**

# Класс "Квадрат"

from Rectangle import Rectangle

from Color import Color

class Square(Rectangle):

FIGURE\_TYPE = "Квадрат"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, a, color):

self.a = a

super().\_\_init\_\_(a, a, color)

def \_\_repr\_\_(self):

return '{} {} цвета со сторонами {} и плащадью {}'.format(

Square.get\_figure\_type(),

self.color.colorprop,

self.a,

self.square()

)

**Файл Circle.py**

# Класс "Круг"

from Color import Color

from Figure import Figure

import math

class Circle(Figure):

FIGURE\_TYPE = "Круг"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, r, color):

self.r = r

self.color = Color()

self.color.colorprop = color

def square(self):

return (self.r \*\* 2) \* math.pi

def \_\_repr\_\_(self):

return '{} {} цвета, с радиусом {} и площадью {}'.format(

Circle.get\_figure\_type(),

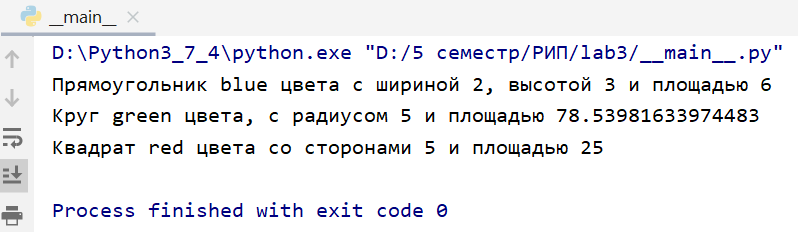
self.color.colorprop,

self.r,

self.square()

)

**Скриншоты**



**Диаграмма классов**

